

代码一共能够分为5个部分的模型基线和一个推理程序。对各模型中各个函数及各部分的说明在代码部分有介绍，这里主要是介绍运行思路。模型融合的主要思路有：1.融合的单一模型应该具有较高的结果。2.模型之间需要保证有足够的差异性。

其中首先运行Model 1 中的 pre-trained-roberta-solution-in-pytorch，该部分以clpr 模型中的预训练模型作为外部数据集，并在该预训练模型的基础上微调(fine-tune)，该模型的最终预测结果为0.467。

Model 2 分为3部分，通过数据集作为任务内的预训练方案(itpt)，模型微调(fine-tune)，从而得到合适的模型结果。model 2首先运行commonlit readable prize roberta torch itpt 预训练程序，然后将得到的结果作为 运行commonlit readable prize roberta torch fit程序微调

其中Model 2使用了两类模型用于训练，一类是使用了Roberta-base模型，另一种是使用了Roberta large模型。其中roberta-base在Model2 的代码中有详细的介绍，能够使用apex，AWP等方法做进一步的优化，本次比赛中的结果为0.477。而roberta-large的两个模型在本次实验中是作为外部数据集存在的，如果有感兴趣的同学可以参考<https://www.kaggle.com/rhtsingh/commonlit-readability-prize-roberta-torch-infer-2> 中的说明对roberta-base中的代码优化得到。两个模型的训练结果均为0.474。三个模型融合的最终结果为0.468 。

Model3 是一个基于tensorflow的模型，该部分模型的训练结果相对其他模型来说是偏低的，该部分模型的主要作用是增大模型之间的差异性，缓解模型过拟合的情况。Model3的分数为0.482。

Model4 运行best-transformer-representations中的MeanPoolingModel模型可获得，该部分的两个模型在kaggle上已经被作者公开(名字分别为model_114和model_72)。这两个模型使用了不同的随机种子，然而model_114的成绩远低于model_72的成绩，甚至两者平均加权的效果都逊色于model_72的成绩。

Model 5 首先运行clrp-pytorch-roberta-pretrain得到模型的预训练模型clpr-pretrain，然后将之前运行clrp-pytorch-roberta-finetune。与其他模型不同的是，Model 5 在推理程序中首先获取模型在测试集中的输出向量，然后将这些向量用SVM预测最终的结果，最后将每个SVM预测的结果平均得到模型最终的预测结果。

最后运行commonlit-ensemble-5x-inference，该部分将所有上述5个模型的预测结果按照一定比例集成在一起，得到最终的预测结果。集成模型能够降低过拟合的风险，使最终结果更加稳定。